Previous Doc   Next Doc   Go to Doc#
First Hit

☐ ▨ Generate Collection ▨

L5: Entry 1 of 17   File: PGPB   Nov 1, 2001

DOCUMENT-IDENTIFIER: US 20010037417 A1
TITLE: Method and system for dynamically dispatching function calls from a first
execution environment to a second execution environment

Application Filing Date:
20010112

Detail Description Paragraph:
[0050] Unlike the prior art in which calls to a different execution environment
with a different binary specification could not be handled in most cases, and in a
limited number of cases could be handled by marshalling the call into a specific
predefined byte stream (for example the CORBA byte stream) for passing to the
different execution environment, call 112 from first execution environment 120 with
a first binary specification is directed to a proxy 130 in a bridge 140. Proxy 130
converts any parameters in the call to parameters for second execution environment
150 using a type description that is described more completely below, and then
dispatches a call 170, with the converted parameters, to service 161 in second
execution environment 150. Call 170 corresponds to call 112 in first execution
environment 120.

Detail Description Paragraph:
[0228] Conversely, if method getRegisteredInterface failed to find a registered
proxy interface, bridge object 405 calls method create proxy with a source
environment and a type as input parameters. In creating a proxy, bridge object 405,
in one embodiment, uses a proxy factory to generate method code to implement each
method specified in the interface to be created. The only information to do this is
a type description of the interface. For example, in a JAVA environment, a binary
class file (*.class) is generated and loaded with the class loader. In the absence
of a loader, which can directly load binary classes, a loader has to be provided.
In a C++ environment, virtual method tables are generated, which delegate each call
to the interface in the source environment.

Detail Description Paragraph:
[0229] The knowledge of the type description is necessary to create the proxy, as
described. This type description is the full description of the limited
functionality, e.g., a description of an interface, in the source execution
environment. The type description may refer one of the different types shown in
Table 5.

Detail Description Paragraph:
[0240] The C++ proxy manages a reference count for the proxy, a pointer to the
bridge of the C++ proxy to obtain the counterpart mapping, the UNO interface the
C++ proxy delegates calls to, the (interface) type the C++ proxy is emulating, and
an object identifier (oid). The type and object identifier are needed to manage
objects from environments, for proof of object identity, and to improve
performance. A proxy to an interface is not needed if there is already a registered
proxy for that interface.

Detail Description Paragraph:

[0269] The UNO proxy manages a reference count for the proxy, a pointer to the bridge of the UNO proxy to obtain the counterpart mapping, the C++ interface the UNO proxy delegates calls to, the (interface) type the UNO proxy is emulating, and an object identifier (oid). The type and object identifier are needed to manage objects from environments, for proof of object identity, and to improve performance. A proxy to an interface is not needed if there is already a registered proxy for that interface.

Detail Description Table CWU:
5 TABLE 5 Environment Type UNO C++ JAVA Byte Signed 8 Bit Signed 8 Bit Signed 8 Bit Short Signed 16 Bit Signed 16 Bit Signed 16 Bit Ushort Unsigned 16 Bit Unsigned 16 Bit Signed 16 Bit Long Signed 32 Bit Signed 32 Bit Signed 32 Bit Ulong Unsigned 32 Bit Unsigned 32 Bit Signed 32 Bit Hyper Signed 64 Bit Signed 64 Bit Signed 64 Bit Uhyper Unsigned 64 Bit Unsigned 64 Bit Signed 64 Bit Float Processor Processor IEEE float dependent: dependent: Intel, Sparc = Intel, Sparc = IEEE float IEEE float Double Processor Processor IEEE double dependent: dependent: Intel, Sparc = Intel, Sparc = IEEE double IEEE double Enum The size of a The size of a All enum values machine word. machine word. of one enum Normally, this Normally, this declaration are is the size of is the size of a static object an integer. an integer. of a class. Each object contains a 32- bit value, which represents the enumeration value. Boolean 1 Byte. 1 Byte. Boolean Char 16 Bit on WNT, 16 Bit on WNT, Unsigned 16 bit W95, W98, and W95, W98, and (char) Os2. 32 Bit on Os2. 32 Bit on Unix Unix String A pointer to a A pointer to a java.lang.String structure which structure which have the have the following following members: members: long refCount; long refCount; long length; long length; wchar_t wchar_t buffer[. . .]; buffer[. . .]; The string in The string in buffer is 0 buffer is 0 terminated. terminated. This is the This is the rtl_wString rtl_wString structure in structure in the rtl-library the rtl-library Structure The structure The structure A class, which contains the contains the is derived from members in the members in the java.lang.Object order of the order of the and contains the declaration. declaration. members in the specified order. Union The size is 4 + The size is 4 + Not specified size of the size of the largest type. largest type. In front of the In front of the union members union members is a long value is a long value (nSelect), (nSelect), which describes which describe the position of the position of the valid the valid member (0 is member (0 is the first). the first). Sequence A pointer to a A pointer to a A normal JAVA structure which structure which array. has the has the following following members: members: void * void * pElements; pElements; long nElements; long nElements; long nRefCount; long nRefCount; The pElements The pElements are a memory are a memory area that area that contains contains nElements nElements elements. elements. Exception Looks like a Looks like a A class, which structure structure is derived from java.lang.Except- ion and contains the members in the specified order. Interface Is a pointer to Is a pointer to A normal JAVA a function a C++-Class interface. table, which which contains at implements least three first the functions. virtual methods queryInterface, acquire and release. Any A structure A structure A class which is that contains a that contains a derived from pointer to a pointer to a "java.lang. type type Object". The description. description. members are a The second The second class, which member is a member is a describe the type pointer to the pointer to the of the value. A value stored in value stored in second member the any. the any. which is the value of the any. Void No memory No memory No memory representation representation representation

<p style="text-align:center"><u>Previous Doc</u>   <u>Next Doc</u>   <u>Go to Doc#</u></p>